SOFTWARE ENGINEERING
(OPEN ELECTIVE – I)

COURSE OBJECTIVE:
1. To understand the software development life cycle.
2. To understand the software requirements and SRS document.
3. To understand the proper design process based on SRS document.
4. To understand the implementation of design and design techniques.
5. To understand software quality, maintenance and reuse.

CORSE OUTCOMES:
1. To understand the process of software engineering and SDLC model.
2. To understand the requirements of the software system process.
3. To understand the different software design processes.
4. To understand the implementation and testing approaches in software engineering.
5. To understand the different quality management, maintenance and reuse policy.

UNIT-I
Software and Software Engineering: Software, characteristics of software, software classification, Software crisis, Software Engineering, Goal of software engineering, Evolution of software engineering methodologies, challenges in Software engineering, software process. **Process model:** Elements of software model, Characteristics of software process, software development life cycle, waterfall model, Iterative waterfall model, Prototype model, Evolutionary model, Spiral model, Agile model of software development (XP model, SCRUM model, RUP)process model

UNIT-II
Requirement analysis and specification: Types of software requirements, requirement engineering process, requirement elicitation, requirement analysis, software requirement specification (SRS), prototyping analysis, requirement specification, requirement validation, formal system specification.

UNIT-III
Software design: Overview of design Process, Characteristics of good software design, principles, modular design, design methodologies, cohesion and coupling, structured design, structured design methodologies, transform vs transaction analysis, Object oriented modeling using UML

UNIT-IV
Implementation: coding principle, coding process, coding standards, coding guidelines, coding verification, code documentation software, coding, code review, software documentation.
Testing: testing fundamental, unit testing, test planning, unit testing, Black Box testing, white box testing, levels of testing, performance testing, usability testing, regression testing, debugging approaches, integration testing

UNIT-V

Software Reliability and quality management: Software Reliability, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model.

Software Maintenance: Software Maintenance, Maintenance Process Model

Software Reuse: What can be reuse? Why almost no reuse so far? Basic issues in Reuse Approach

TEXT BOOKS:
1. Software Engineering, A Practitioner's Approach, Roger S. Pressman, Seventh Edition, McGrawHill International Edition
2. Fundamental of Software Engineering, Rajib Mall, Third Edition, PHI
3. Software Engineering, Ian Sommerville, Ninth Edition, Pearson Education

REFERENCE BOOKS:
1. Fundamental of Software Engineering, Rajib Mall, Third Edition, PHI
2. Software Engineering : A primer, Waman S Jawadekar, Tata McGrawHill, 2008
3. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010
4. Software Engineering: Abstraction and Modeling, Dinner Bjorner, Springer International edition